

## ASSIGNMENT - 4

Date	11 November 2022
Team ID	PNT2022TMID32816
Project Name	SMARTFARMER - IoT enabled smart farming applications
Student Name	UBENDRAN.V

### QUESTION:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 CMS send "alert" to IBM cloud and display in device recent events.

### CODE:

```
#include <WiFi.h>                // library for WIFI

#include <PubSubClient.h>         // library for MQTT

//----- credentials of IBM Accounts -----

#define ORG " eneg3n "
#define DEVICE_TYPE " RASPBERRYPI "
#define DEVICE_ID " 80722 "
#define TOKEN " 8072251864 "
#define speed 0.034
#define led 14
String data3;
int LED = 4;

//----- customise above values -----

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/tukadu /fmt/json";             // topic name and type of
event perform and
format in which data to be send
```

```
char topic[] = "iot-2/cmd/led/fmt/String";          // cmd Represent type and command is
test format of
strings
```

```
char authMethod[] = "use-token-auth";              // authentication method char
```

```
token[] = TOKEN;
```

```
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //Client id
```

```
//-----
```

```
WiFiClient wifiClient;          // creating instance for wificlient
```

```
PubSubClient client(server, 1883, wifiClient); // calling the predefined client id by passing
parameter like server id,port and wifi credential
```

```
const int trigpin=5; const
```

```
int echopin=18;
```

```
String command;
```

```
String data="";
```

```
long duration; float
```

```
dist;
```

```
void setup()
```

```
{
```

```
Serial.begin(115200)
```

```
;      pinMode(led,
OUTPUT);
```

```
pinMode(trigpin,
OUTPUT);
```

```
pinMode(echopin,
INPUT);
```

```
wifiConnect();
```

```
mqttConnect();
```

```
}
```

```
void loop() { bool isNearby
```

```

= dist < 100;
digitalWrite(led, isNearby);

publishData();
delay(500);

if (!client.loop())
{
  mqttConnect();          // function call to connect to ibm
}
}

/* -----retrieving to cloud-----
-----*/

void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST",
  "", 6); while (WiFi.status() !=
  WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

void mqttConnect()
{
  if (!client.connected())
  {
    Serial.print("Reconnecting MQTT client to
    "); Serial.println(server); while

```

```

(!client.connect(clientId, authMethod,
token))
{
  Serial.print(".");
  delay(500);
}
initManagedDevice();
Serial.println();
}
}

void initManagedDevice() {
  if (client.subscribe(topic))
  {
    Serial.println("IBM subscribe to cmd OK");
  }
  else
  {
    Serial.println("subscribe to cmd FAILED");
  }
}

void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  if(dist<100)
  {
    digitalWrite(LED,HIGH); String
    payload = "{\"Alert Distance\":\"";
    payload += dist;
    payload += "\"}";
  }
}

```

```

Serial.print("\n");
Serial.print("Sending payload: "); Serial.println(payload); if (client.publish(publishTopic,
(char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok else
prints publish failed {
Serial.println("Publish OK");
}
}
if(dist>100)
{
digitalWrite(LED,HIGH);
String payload =
"{\"Distance\":\""; payload
+= dist;
payload += "\"}";

Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
Serial.println("Publish OK");
}
else
{
digitalWrite(LED,LOW);
Serial.println("Publish FAILED");
}

}
}

```

**OUTPUT :**

## Code simulation on wokwi

The screenshot shows the Wokwi IDE interface with the following components:

- Code Editor:** Displays the `esp32-dht22.ino` file. The code includes headers for `WiFi` and `PubSubClient`, defines IBM Cloud credentials (ORG, DEVICE\_TYPE, DEVICE\_ID, TOKEN), and sets up a WiFi client and MQTT publisher. It publishes distance data to the topic `iot-2/cmd/led/fmt/String`.
- Simulation Window:** Shows a visual representation of the ESP32-DHT22 sensor module connected to a breadboard with an LED and a buzzer.
- Console Output:** Displays the simulation logs, including "Connecting to Wifi..Wifi connected, IP address: 10.10.0.2" and "Reconnecting MQTT client to rwazv5.messaging.internetofthings.ibmcloud.com".

## Data sent to IBM Cloud with distance

The screenshot shows the IBM Cloud IoT Gateway dashboard with the following details:

- Navigation:** Includes tabs for "Browse", "Action", "Device Types", and "Interfaces". A sidebar on the left contains icons for various dashboard functions.
- Device Selection:** A dropdown menu shows the selected device as "distance".
- Recent Events:** A table displays the live stream of data coming from the device. The table has columns for "Event", "Value", "Format", and "Last Received".
- Simulation Status:** A status bar at the bottom indicates "1 Simulation running".

Event	Value	Format	Last Received
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago